

GRAPE 2-D Grid Generator for Turbomachinery

User's Manual and Documentation

Version 107, May 2011

Dr. Rodrick V. Chima
NASA Glenn Research Center, MS 5-12
21000 Brookpark Road
Cleveland, Ohio 44135 USA

phone: 216-433-5919
email: chima@nasa.gov
web: <http://www.grc.nasa.gov/WWW/5810/rvc>
download: <http://sr.grc.nasa.gov>

Introduction

The GRAPE (GRids about Airfoils using Poisson's Equation) code is an elliptic grid generator originally intended for isolated airfoils, as shown in Fig. 1. The code was developed by Reece Sorenson at NASA Ames Research Center [1, 2]. Reference [1] describes the theory behind the original GRAPE code, and reference [2] is a detailed users' manual for external flow problems. Any publications resulting from the use of this code should include these two references.

GRAPE was modified by the author to allow generation of periodic C-type grids for turbomachinery blades. Figures 2 – 4 show examples of grids for a centrifugal compressor rotor, a turbine vane, and an axial compressor rotor. Input files for these three grids are included with GRAPE as sample cases. Turbomachinery grids generated with GRAPE can be used directly with the RVCQ3D turbomachinery analysis code [3 - 4.] RVCQ3D is distributed with input for the same sample cases that are included with GRAPE.

This document describes the modifications made to the GRAPE code to allow generation of periodic C-type grids for turbomachinery blades, and serves as the users' manual. Instructions for unzipping, compiling, and running the code are included. Input variables used for turbomachinery grids are described in full. Other variables used for isolated airfoils or general grids are listed and described briefly, but the user should consult Sorenson's original users' manual for more information [2].

GRAPE allows arbitrary specification of inner and outer boundary points, generates interior points algebraically, and then smoothes those points by solving a Poisson equation. Forcing terms in the Poisson equation are chosen to maintain the desired grid spacing and intersection angles at the inner and outer boundaries.

GRAPE is written in a Cartesian (x, y) coordinate system that is mapped to a general body-fitted (ξ, η) coordinate system. Turbomachinery blades, however, are often specified on a surface of revolution in a cylindrical $(m, \bar{r}\theta)$ system, where m is the arc length along the surface and θ is the circumferential direction. To use GRAPE for turbomachinery blades, θ -coordinates are multiplied by a mean radius \bar{r} , so that m and $\bar{r}\theta$ both have units of length. Blade coordinates can be scaled in advance, or done within the GRAPE code using the variable *yscl*, described later. Thus, blade coordinates are input as $(m, \bar{r}\theta)$ pairs, but GRAPE treats them internally as (x, y) .

Many modifications were made to the original GRAPE code, but all input options described in the original code [2] have been retained. New inner and outer boundary routines were added for turbomachinery blades. The new inner boundary routine adds several input parameters that give considerable control over the spacing of points on the blade surface. The new outer boundary routine generates periodic boundaries automatically from the blade mean camber line. Outer boundaries are generated by shifting the mean camber line one-half pitch up and down. The mean camber lines are then extended up- and downstream using polynomial curves. Points can be distributed with arc-length along the outer boundary, and are aligned to be periodic top-to-bottom.

GRAPE is written completely in Fortran and runs in seconds on a Linux workstation. GRAPE will run on a PC or Mac, but the user will have to make the appropriate conversions. Code input is supplied as a text file with grid

parameters specified using namelist input, and blade shapes input as (x, y) pairs. Some printed output is provided. No graphical output is provided, but grid files can be read directly and plotted using the public domain CFD visualization code PLOT3D, or the commercial codes FieldView and Tecplot.

This documentation describes how to dimension, compile, and run the GRAPE code on a Linux workstation. The output file format is described. Finally all namelist input variables and blade coordinate input options are described in detail.

References

1. Steger, J. L., and Sorenson, R. L. "Automatic Mesh Point Clustering Near A Boundary in Grid Generation with Elliptic Partial Differential Equations," *Journal of Computational Physics*, Vol. 33, No. 3, Dec. 1979, pp.405-410.
2. Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by Use of Poisson's Equation," NASA TM-81198, 1980.
3. Chima, Rodrick V., "Explicit Multigrid Algorithm for Quasi-Three-Dimensional Viscous Flows in Turbomachinery," *AIAA Journal of Propulsion and Power*, Vol. 3, No. 5, Sept.-Oct. 1987, pp. 397-405.
4. Chima, Rodrick V., "A $k-\omega$ Turbulence Model for Quasi-Three-Dimensional Turbomachinery Flows," AIAA Paper 96- 0248, Jan. 1995.

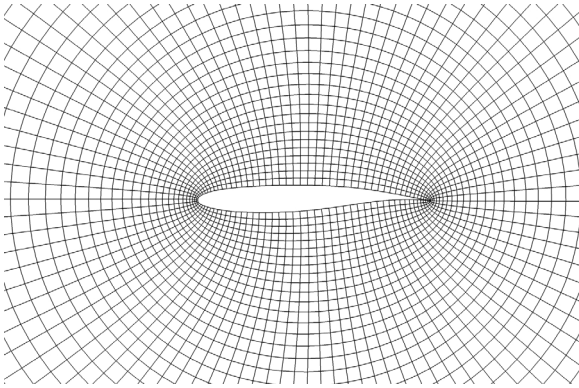


Figure 1. O-grid around a Garabedian-Korn airfoil.

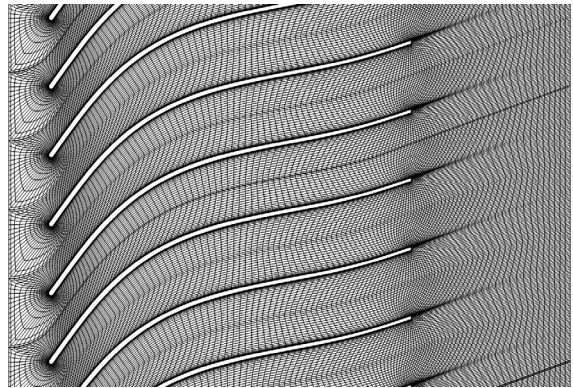


Figure 2. C-grid for a centrifugal compressor blade.

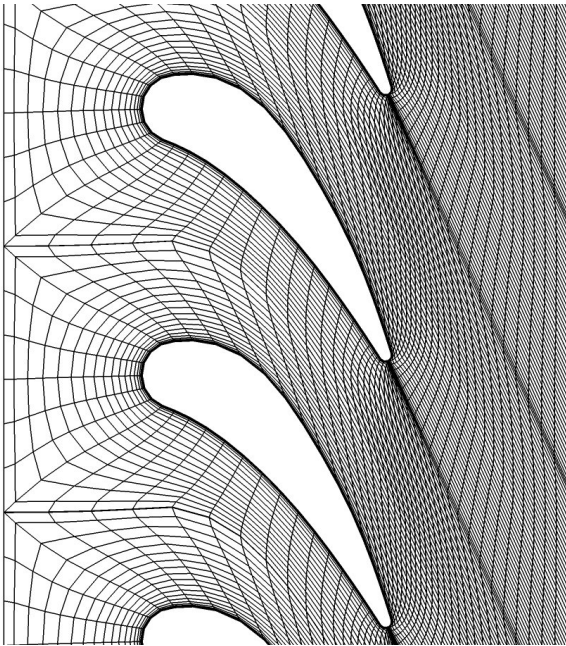


Figure 3. C-grid for a turbine vane.



Figure 4. C-grid for rotor 37 at 70 percent span.

Unzipping, Compiling, and Running GRAPE

GRAPE is supplied as a zipped file. It will unzip into a directory with the same name as the file. This documentation should be in the main directory. There are subdirectories for source code and test cases. On a Linux system:

```
unzip grape_107.zip
```

Compiling GRAPE

Go to the src directory and edit the Makefile. Compiler commands are set for the Intel Linux compiler,
FC = ifort -O3

Change the commands as necessary for your compiler. Full optimization usually works well. Near the bottom of the Makefile there may be a line that moves the executable to a bin directory. Keep or remove this line as desired.

```
mv grape ~/bin
```

Save the file and exit. Edit gridp and modify maximum grid dimensions if desired.

```
parameter (idm=385, jdm=74)
```

Run make. Move the executable to a directory in your path.

Clean up object and executable files if desired by running

```
make clean
```

Running GRAPE

GRAPE is run as a standard Linux process:

```
grape < input_file > output_file
```

The output file is not too useful so you may want to ignore the "> output_file" and watch the output scroll up the screen. The last line of output prints variables *mtl* and *mil* used by RVCQ3D. You may want to make note of these.

Output Grid File

The output grid file is written to Fortran unit 1 (fort.1). The file is an unformatted binary file, which can be linked to a file name before running GRAPE,

```
ln file.xyz fort.1
```

or renamed after running GRAPE,

```
mv fort.1 file.xyz
```

Binary grid files can be used immediately by RVCQ3D on the same type of computer on which they were generated. Grids are stored using standard PLOT3D xyz-file structure. Examine the grid with PLOT3D, FieldView, or TecPlot using the xyz, 2-D, and unformatted file options.

Grids can be read with the following Fortran code:

```
!      read grid coordinates
      read(1) jmax, kmax
      read(1)
      &((x(j,k), j=1, jmax), k=1, kmax), ((y(j,k), j=1, jmax), k=1, kmax)
```

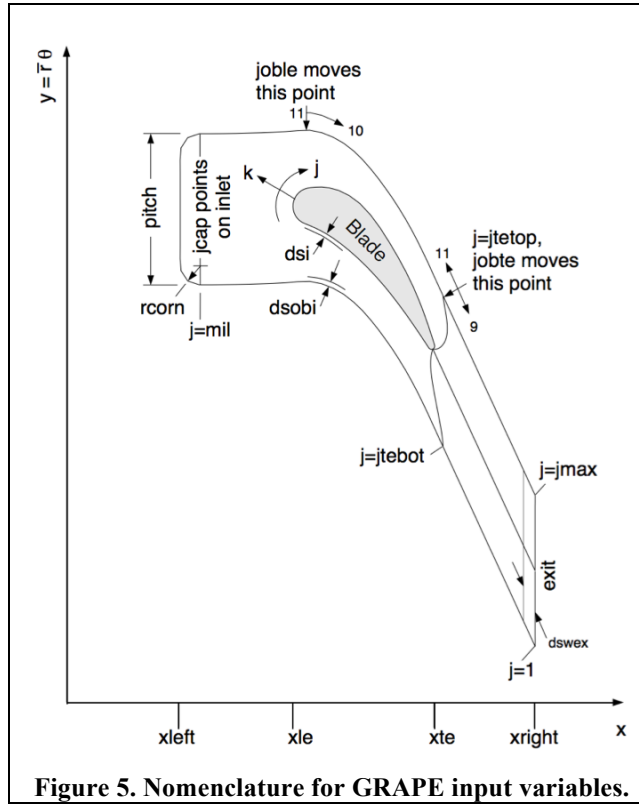


Figure 5. Nomenclature for GRAPE input variables.

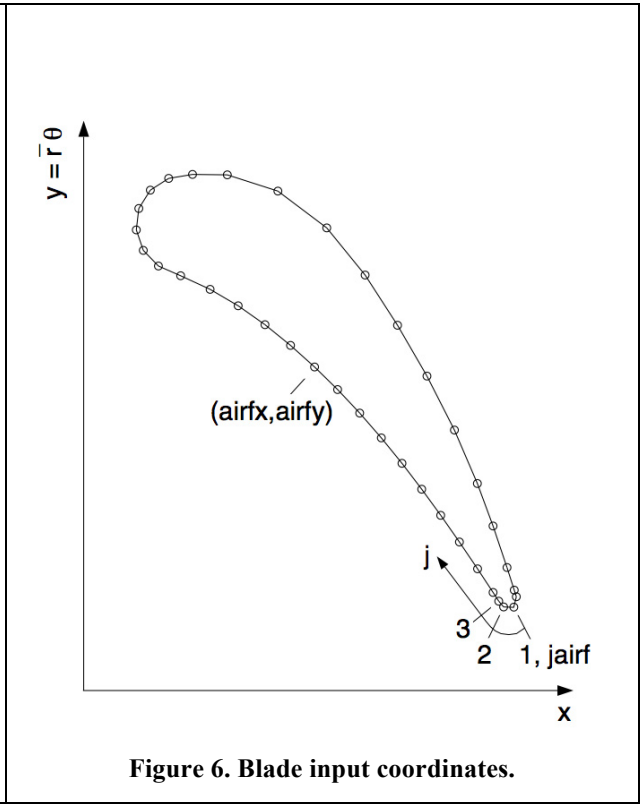


Figure 6. Blade input coordinates.

Namelist Input

This section describes the input parameters that are commonly used for turbomachinery blades. Only C-type grids are supported for turbomachinery. Figures 2 - 4 show examples of the C-type grids distributed as test cases with GRAPE and RVCQ3D.

GRAPE supports many options for generating arbitrary inner and outer boundaries for isolated airfoils and other geometries. Input variables used for those options are described briefly at the end of each section, but the user is referred to the original GRAPE documentation [2] for more information.

GRAPE input consists of a text file with three blocks of namelist input. The first two blocks include parameters controlling grid size, inner and outer boundary stretching, and operation of the elliptic smoother. The third block contains blade coordinates entered as (x, y) pairs. Many of the input parameters are illustrated in Figs. 5 and 6.

Default values are set in a block data routine. NOTE: Many of the defaults are Sorenson's original values chosen for isolated airfoils and must be reset for turbomachinery problems. In the input description below, default values are given in angle brackets, <default=value> or <default.> If no default is given the value must be input.

grid1 - Grid Size and Outer Boundary Parameters

<i>jmax</i>	Grid size in the j- (streamwise) direction (see Fig. 5.) Typically 128-256. <default=100>
<i>kmax</i>	Grid size in the k- (blade-to-blade) direction (see Fig. 5.) Typically 35 (inviscid) to 55 (viscous). <default=49>
<i>jtebot</i>	j-index of the lower trailing-edge point on a C-grid (see Fig. 5.) <default=15>
<i>jtetop</i>	j-index of upper trailing-edge point on a C-grid. Older versions of GRAPE required that <i>jtetop</i> be periodic with <i>jtebot</i> , i.e., $jtetop = jmax + 1 - jtebot$, but this is now set internally and <i>jtetop</i> can be omitted.

<i>ntetyp</i>	MUST = 3 for turbomachinery problems. <default=1> (Controls type of grid, O or C, in the original code.)
<i>nairf</i>	MUST = 5 for turbomachinery problems. <default=2> (Controls inner boundary spacing in the original code.)
<i>nobshp</i>	MUST = 7 for turbomachinery problems. <default=1> (Controls type of outer boundary in the original code.)
<i>jairf</i>	Number of blade input points entered in variables <i>airfx</i> and <i>airfy</i> in grid3 (see Fig. 5.) <default=0>
<i>nibdst</i>	Flag for type of clustering along the blade surfaces, span, and wake, <default=1> = 6 Hyperbolic tangent clustering - smoothest, but may be sparse at blade center if <i>jmax</i> is small. <default> = 7 Hermite polynomial clustering - more uniform, but may grow too quickly near leading and trailing edges. Good for large <i>jmax</i> .
<i>dsi</i>	Grid spacing away from the inner boundary, in same units as blade input (see Fig. 5.) <default=0.01> For inviscid solutions use $dsi \approx O(\text{chord}/100)$, or choose <i>dsi</i> to give square cells around the leading or trailing edges. For viscous solutions use $dsi \approx \text{chord}/10,000$. Choose larger values for quick solutions, smaller values for accurate loss, skin friction, or heat transfer.
<i>xleft</i>	x-coordinate of the inlet boundary of the grid (see Fig. 5.) <default=-6> Typically $xleft = xle - \text{pitch}/2$. Smaller values give a more uniform grid; larger values give a stretched grid.
<i>xright</i>	x-coordinate of the exit boundary of the grid (see Fig. 5.) <default=6> C-type grids can extend indefinitely downstream as long as <i>jtebot</i> is large enough to resolve the downstream region.
<i>xle</i>	x-coordinate of the blade leading edge (see Fig. 5.) <default=0> The input airfoil coordinates are rescaled to go from <i>xle</i> to <i>xte</i> . To prevent rescaling, set <i>xle</i> to the minimum value of the input blade coordinates.
<i>xte</i>	x-coordinate of the blade trailing edge, (see Fig. 5.) <default=1> The input airfoil coordinates are rescaled to go from <i>xle</i> to <i>xte</i> . To prevent rescaling, set <i>xte</i> to the maximum value of the input blade coordinates.
<i>rcorn</i>	Radius for the front corner of the C-grid (see Fig. 5.) <default=1> Use <i>rcorn</i> = 0.0 to give square corners.
<i>maxita</i> (2)	Array with two elements: maximum number of iterations on each grid level. <default=200, 100> GRAPE supports multigrid convergence acceleration, but it is so fast on modern computers that multigrid is not needed. The first element of <i>maxita</i> refers to coarse grids and the second element refers to the finest grid. Generally <i>maxita</i> (1) = 0 to skip the coarse grids and <i>maxita</i> (2) = 100 to 300. Use <i>maxita</i> = 0, 0 to see the initial grid if the Poisson solver fails.
<i>norda</i> (2)	Array with two elements: number of orders of magnitude to reduce the residuals for convergence on each grid level, <default=4, 1> The first element refers to coarse grids and the second element refers to the finest grid. Generally <i>norda</i> (1) = 0 to skip the coarse grids and <i>norda</i> (2) = 3 for three orders of magnitude reduction in the residual. GRAPE will usually terminate after <i>maxita</i> iterations before the <i>norda</i> limit is reached.

nout MUST = 4 for turbomachinery problems, <default=1>
Parameter controlling output options. *nout*=4 writes a binary grid to unit 1 in PLOT3D format.

Other Variables in grid1 (See Ref. 2 for details)

tr Thickness ratio for NACA airfoils.

radob Outer boundary radius for O-grids.

ytop, ybotom y-coordinates used for an obsolete cascade outer boundary option (*nobshp* = 5).

alamf, alarmr Left and right boundary angles for an obsolete cascade outer boundary option (*nobshp* = 5).

jpri Controls output printing options.

grid2 - Grid Spacing and Algorithm Parameters

nobcas Controls angle between η -grid lines and periodic (outer) boundary. This was an undocumented feature in the original GRAPE code.
= 0 η -lines are vertical at the periodic boundary. <default, recommended>
= 1 η -lines are normal to the periodic boundary.

pitch Cascade pitch in same units as blade input (see Fig. 5.) <default=1>

yscl Scale factor for blade input y-coordinates. <default=1>
For an annular cascade the y-coordinates are input as $\bar{r}\theta$. If θ -coordinates are input for *yairf*, use *yscl* = \bar{r} to rescale the input. Note that x-coordinates can be rescaled using *xle* and *xte*.

dsobi Grid spacing away from the periodic boundary, in same units as blade input (see Fig. 5). <default=0.2>
Use *dsobi* \approx *pitch* / *k* max to get nearly square spacing at the periodic boundary.

jcapi Number of j-points on the inlet part of the C-grid (see Fig. 5).
Remaining points are distributed over the periodic boundaries. Increase *jcapi* to pull points towards the inlet, and vice-versa.

nle Number of points equally-spaced around the blade leading edge. <default=15>

nte Number of points equally-spaced around the blade trailing edge, half on top, half on the bottom. Should be an even number. <default=10>

dsle Spacing around the leading edge as a fraction of total arc length around blade. <default=.0025>

dste Spacing around the trailing edge as a fraction of total arc length around blade. <default=.0025>

xtfrac Parameter that controls x-spacing away from the trailing edge. <default=1>
The x spacing away from the trailing edge is roughly *xtfrac* \times *dste*. Start with *xtfrac* = 1 and adjust if necessary.

dswex x-spacing at the grid exit, (see Fig. 5.) <default is equally-spaced in x.>
Only used if *jwakex* = 1 to stretch grid the downstream.
The grid spacing along the wake cut stretches from *xtfrac* \times *dste* at the trailing edge, to *dswex* at the exit. *Dswex* should be roughly $(x_{right} - x_{te}) / j_{tebot}$.

dsra (Pressure surface arc length)/(total surface arc length). <default=0.5>

Used to locate the center of the leading edge clustering on the blade. The clustering is centered about $dsra \times (\text{total surface arc length})$. Typical values are 0.5 for symmetrical blades, about 0.49 for compressor blades, and about 0.45 for highly cambered turbine blades. Start with the typical value and adjust in small increments until the equally spaced leading edge clustering is centered on the leading edge of the blade.

- joble* The periodic outer boundary for a C-grid is made up of three segments, a polynomial segment upstream, the mean-camber line between the blades, and a linear segment downstream. Within GRAPE each segment is represented by an array of 10 or 11 points that are later reclustered. *Joble* is the index where the upstream segment connects to the mean-camber line. It can be used to manipulate the shape of the outer boundary to a limited extent (see Fig. 5). Values can be 11, 10, 9, etc. The default <11> starts the upstream quadratic segment at the leading edge. Smaller values move the starting point inside the passage about 10 percent chord per value.
- fswake* Fractional distance along the downstream periodic boundary where the k-grid line from the trailing edge intersects the outer boundary. The default value of <1.0> places the outer boundary points at *xte*. On some blades this can cause the k-lines to cross the trailing edge. In this case try setting *fswake* < 1.0 to pull the grid lines towards the downstream boundary.
- jwakex* Flag for stretching the outer boundary grid spacing along the wake (j-direction).
 = 0 Equally-spaced outer boundary along the wake.
 = 1 Stretched outer boundary along the wake. Spacing at the trailing edge is set by $xtfrac \times dste$, spacing at the exit is set by *dswex*. <default>
- kwakex* Flag for expanding the grid spacing across the wake (k-direction).
 = 0 Grid is clustered along the wake cut. Gives better wake resolution if the wake follows the cut. <default>
 = 1 Grid expands away from the wake cut moving downstream. Gives a more uniform grid downstream.
- aaai, bbbi* Exponents controlling how far angles and spacings specified at the inner boundary propagate into the interior. Small *aaai* and *bbbi* give large distances but slow convergence, and vice versa. Any value > 0.0 is acceptable. See [2] for details. Typically 0.25 - 0.65. < 0.45>
- ccci, dddi* Like *aaai* and *bbbi*, but for the outer boundary. Typically 0.25 - 0.65. < 0.45>
- cs moo* Smoothing coefficient for periodic boundary. < 0.0>
 Normally the outer (periodic) boundary points are fixed. Setting *cs moo* = (0.1 to 1.0) allows the outer boundary points to float by smoothing them with a Laplace-type filter after each iteration of the elliptic solver. This may improve the grid while sacrificing control of angles at the outer boundary. Non-zero values of *cs moo* may prevent the GRAPE code from converging, although the grids may still be useful.
- exl, exr* Exponent controlling the shape of the left (upstream) and right (downstream) boundaries of the grid. The boundaries start tangent to the mean camber line and then curve to axial at rates determined by *exl* and *exr*.
 > 10 No curvature – the boundary is a linear extension of the mean camber line.
 > 1.5 Slow curvature to axial.
 = 1.5 Moderate curvature to axial <default>
 < 1.5 Fast curvature to axial.
 = 1.0 Turns the boundary abruptly to axial.

Other Variables in grid2 (See Ref. 2 for details)

- nletyp* Controls leading edge shape, sharp or blunt. <1 = blunt>

<i>binn</i>	Controls point spacing on the airfoil for airfoil grids. <1.1>
<i>wakep</i>	Controls point spacing on the wake cut for airfoil grids. <1.0>
<i>jdists</i>	Number of points in used-supplied distribution function (see <i>dist</i> in grid3.) <0>
<i>jcamb</i>	Number of points in used-supplied camber functions (see <i>cambx</i> , <i>camb</i> in grid3.) <0>
<i>rotang</i> , <i>rotctr</i>	Airfoil rotation angle and center of rotation. <0.0, 0.0>
<i>xobcnt</i>	x-location of center of O-grid when <i>nobshp</i> =1 <0.0>
<i>omega</i>	SLOR relaxation factor for the Poisson solver. <1.3>
<i>omegp</i> , <i>omegq</i>	Relaxation factors for the inner boundary. Turn off inner boundary control by setting to 0.0 <0.3>
<i>omegr</i> , <i>omegs</i>	Relaxation factors for the outer boundary. Turn off outer boundary control by setting to 0.0 <0.3>
<i>plim</i> , <i>qlim</i>	Limiters for inner boundary forcing functions. <1.0>
<i>rlim</i> , <i>slim</i>	Limiters for outer boundary forcing functions. <1.0>
<i>thetai</i> , <i>thobi</i>	Grid line intersection angle at inner and outer boundaries <90.0>
<i>teopen</i>	Distance across the open trailing edge of an airfoil. <0.0>

grid3 – Blade Coordinates

<i>airfx</i>	Array of x- or m-coordinates around the blade, starting at the trailing edge, going clockwise around the blade, and repeating the first point (see Fig. 6.) The coordinates are fit with a monotone cubic spline and must adequately define the leading and trailing edges.
<i>airfy</i>	Array of y- or $\bar{r}\theta$ – coordinates around the blade, ordered like <i>airfx</i> , (see Fig. 6).

Other Variables in grid3 (See Ref. 2 for details)

<i>ds</i> , <i>dsob</i>	Arrays of <i>jmax</i> inner and outer boundary spacings.
<i>theta</i> , <i>thetob</i>	Arrays of <i>jmax</i> inner and outer boundary angles.
<i>dist</i>	Array of <i>jdists</i> values of the airfoil point distribution function.
<i>cambx</i> , <i>camb</i>	Arrays of <i>jcamb</i> values of a camber distribution function.
<i>obangs</i>	Array of <i>jmax</i> outer boundary angles.
<i>xob</i> , <i>yob</i>	Arrays of <i>jmax</i> (x,y) coordinates of the outer boundary.
<i>aaa</i> , <i>bbb</i> <i>ccc</i> , <i>ddd</i>	Arrays of <i>jmax</i> values of boundary control exponents.